САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Филиппова Анна Михайловна

Выпускная квалификационная работа бакалавра

«Контроль заполнения медицинской документации на основе извлечения и применения ассоциативных правил»

Направление 010400

Прикладная математика и информатика

Научный руководитель,
кандидат физ.-мат. наук,
доцент Добрынин В.Ю.

Санкт-Петербург

2017

Содержание

Введение

Анализ данных является активно развивающейся областью в математике и информатике, что связано с тем, что в настоящее время через компании, которые занимаются разработкой, внедрением и дальнейшим сопровождением программного обеспечения для учреждений здравоохранения, проходит колоссальное количество информации, на обработку и проверку которой сотрудникам приходится тратить много времени [1]. Таким образом, автоматизация процесса проверки является одним из приоритетных направлений развития служб технической поддержки.

Проблема отсутствия автоматизации анализа объемных баз данных явно проявляется при работе с медицинской документацией, правильность заполнения которой играет важную роль. Допущенные в силу человеческого фактора ошибки ведут к тому, что истории болезней, которые включаются в счета и передаются в страховые компании на проверку, возвращаются обратно без оплаты, что порождает недопонимание между медицинскими учреждениями и страховыми компаниями. С целью понизить процент возвращаемых счетов, данные об историях болезней предварительно проверяются сотрудниками технической поддержки, что является длительным и трудозатратным процессом.

Решением данной проблемы может стать использование алгоритмов поиска ассоциативных правил. В связи с этим, целью данного дипломного исследования является построение ассоциативных правил для обнаружения ошибок в медицинской документации и создание программного обеспечения, реализующего контроль правильности заполнения на основе выявленных закономерностей. Данная работа посвящена рассмотрению и анализу вариантов ошибок, которые совершаются при внесении данных в истории болезней пациентов.

Постановка задачи

Учитывая актуальность проблемы автоматизации процесса проверки заполнения электронных медицинских карточек, были поставлены следующие задачи.

Во-первых, изучить структуру медицинской информационной системы (далее - МИС) компании «Виста» и предоставленную ею базу данных. Рассмотреть варианты ошибок, допускаемых специалистами при заполнении электронных карточек пациентов, а именно, проанализировать справочник причин возврата счетов. После чего составить список полей, включаемых в счета для страховых компаний и имеющих в себе наиболее часто-встречающиеся ошибки. Составить обучающую выборку из базы данных для нахождения в ней зависимостей.

Во-вторых, рассмотреть понятие построения ассоциативных правил для выявления закономерностей среди характеристических данных (таких как пол, возраст, диагноз, услуги), входящих в истории болезней и далее включаемых в счета. А так же изучить алгоритм поиска закономерностей - Apriori.

В-третьих, составить тестовую выборку, содержащую в себе набор карточек с наличием ошибок. Написать программу, которая, проходя по контрольной выборке, выявляет те карточки, в которых были предварительно допущены ошибки. Найти оптимальные параметры для алгоритма нахождения АП, повышая процент найденных неправильных карточек.

Обзор литературы

В статье «Mining association rules between sets of items in large databases» трех авторов R. Agrawal, T. Imielinski и A. Swami в 1993 году была представлена задача нахождения ассоциативных правил в большой базе данных транзакций [2]. Решение этой проблемы было позже описано в технической статье «Fast algorithms for Mining Association rules in large database» R.Srikant, R. Agrawal, где авторы познакомили читателей с алгоритмом Apriori, который позволяет быстро генерировать наборы товаров для поиска в них зависимостей [3]. В небольшой статье «The arules R-Package Ecosystem: Analyzing Interesting Patterns from Large Transaction Data Sets» описывается вводная часть в R-пакеты, которые позволяют создавать базы транзакций и управлять алгоритмами эффективного поиска и анализа ассоциативных правил [4]. Подробное описание пакета arules приводится в «arules - A Computational Environment for Mining Association Rules and Frequent Item Sets», где расписываются значение и применения его классов [5]. На выходе функция apriori пакета arules выводит найденные правила и соответствующие им меры: support, confidence и lift. Мера лифт была описана в статье «Dynamic Itemset Counting and Implication Rules for Market Basket Data» в 4 пункте «Implication rules», где она была представлена как мера интереса [6]. Про способы визуализации с примерами написано в технической статье «Visualizing Association Rules: Introduction to the R-extension Package arulesViz» [7]. В работе используется функция sample(), которая сокращает огромную базу данных до случайной выборки из n элементов, при этом не теряя правила, которые могли быть найдены во входных данных. Вывод формулы для нахождения нужного размера n выборки приводится в статье «Evaluation of Sampling for Data Mining of Association Rules» M. Zaki, S. Parthasarathy, W Li, M. Ogihara [8].

Глава 1. Подготовка данных

У каждого гражданина Российской Федерации есть полис обязательного медицинского страхования (далее - ОМС), по которому медицинские организации обязаны оказывать бесплатную помощь в рамках программ обязательного медицинского страхования [9]. В фонд ОМС каждый месяц отправляются отчеты о предоставленных медицинских услугах в лечебно-профилактических учреждениях (далее - ЛПУ). После проверки фонд ОМС перечисляет ЛПУ деньги, которые перераспределяются, согласно нуждам учреждений [10]. При обнаружении фондом ОМС ошибок, счета возвращаются без оплаты. Для анализа счетов на наличие ошибок возможно написание прикладного программного обеспечения, для создания которого необходимо иметь четкое представление о медицинской информационной системе, о частых ошибках, которые включены в справочник причин возврата счетов, о структуре базы данных, хранящей истории болезней, а также требуется иметь список полей, проверка которых будет производиться при помощи поиска ассоциативных правил.

1.1 Разбор МИС «Виста-мед»

Одним из направлений работы компании ООО «Виста» является техническая поддержка и сопровождение своих продуктов, внедренных в ЛПУ. Разработанная для учреждений здравоохранения, МИС КПС «Виста-мед» решает проблему бумажного документооборота, длинных очередей в регистратуру, экономии времени как врачей, так и пациентов.

МИС включает в себя ведение электронной истории болезни пациента, где хранятся все данные о его визитах и заключениях врачей [11]. На основе введенных данных составляются счета, которые предоставляются страховым компаниям. Так как ввод данных осуществляется вручную, при заполнении электронной карты врачами в силу человеческого фактора совершаются

ошибки, на обнаружение которых тратится много времени. После проверки карт сотрудниками страховых компаний, большая часть счетов возвращается без финансирования, что приводит к недопониманию между медицинскими специалистами и разработчиками программного обеспечения.

С целью соответствовать интересам потребителей, в программу был встроен алгоритм проверки наличия явных ошибок, указывающий, например, на отсутствие указания фамилии пациента или номера его полиса. Вдобавок счета перед отправкой предварительно проверяются сотрудниками отдела автоматизированной системы управления технологическим процессом (далее - АСУ ТП), что является достаточно ресурсоемкой задачей. Тем ни менее, в силу присутствия человеческого фактора проверка документации сотрудником также не является гарантией правильности ее заполнения.

1.2 Составление счетов и справочник причин их возвратов

В счета, предоставляемые страховым компаниям, включаются лишь некоторые поля базы данных. Для выявления нужных был проведен анализ ошибок на основе справочника причин возврата счетов [12]. Данный справочник представляет собой набор из 437 ошибок, допускаемых специалистами в заполнении историй болезней. Данные ошибки можно примерно разделить на две категории: когда поле не заполнено и когда заполненное поле не соответствует другим характеристикам карточки пациента. В первом случае обнаружить ошибку можно простым запросом в базе данных, а во втором - при помощи сопоставления с найденными ассоциативными правилами. В таблице приведены некоторые из ошибок, исправить которые можно, основываясь на поиске закономерностей в базе данных.

Таблица 1. Примеры ошибок из справочника "Причины возврата счетов"

Код	Наименование ошибки
315	Несоответствие оказанной мед. услуги и специальности медицинского сотрудника
325	Не совпадает код услуги исполнителя с направлением
507	Профиль ОМП (или специальность врача) не соответствует диагнозу
508	Профиль ОМП (или специальность врача) не соответствуют возрасту пациента
533	Профиль кода услуги не соответствует диагнозу (непрофильная госпитализация или амбулаторный прием)
537	Диагноз не соответствует возрасту (или полу) пациента
733	Профиль кода услуги не соответствует диагнозу (непрофильная госпитализация или амбулаторный прием)
351	Значение в поле "Пол" не соответствует диагнозу
360	Оказаны услуги с видом помощи детская поликлиника или детская стоматология взрослому человеку
361	Мужчине в гинекологии оказаны медицинские услуги
362	В урологии оказаны медицинские услуги женщине с диагнозами N40-N51
570	Диагноз не соответствует состоянию пациента
861	Оказанная медицинская услуга не соответствует полу пациента
574	Некорректно выставлена специальность врача (или профиль ОМП)
888	Диспансерный прием врача-стоматолога детского (с возраста 3 лет) оказан не по возрасту
889	Диспансерный прием детского врача эндокринолога (с возраста 5 лет) оказан не по возрасту
894	Электрокардиография проведена не по возрасту
897	Проведено не по возрасту УЗИ щитовидной железы и органов репродуктивной сферы (с возраста 7 лет)
901	Диспансеризация детей-сирот оказана гражданину старше 18 лет
906	Оказан профилактический осмотр ребенку в возрасте 3 лет и старше
910	Несоответствие диагноза исходу заболевания или исходу обращения
911	Несоответствие профиля оказанной МП и специальности мед. сотрудника
930	Диагноз не соответствует возрасту пациента
931	Диагноз не соответствует специальности

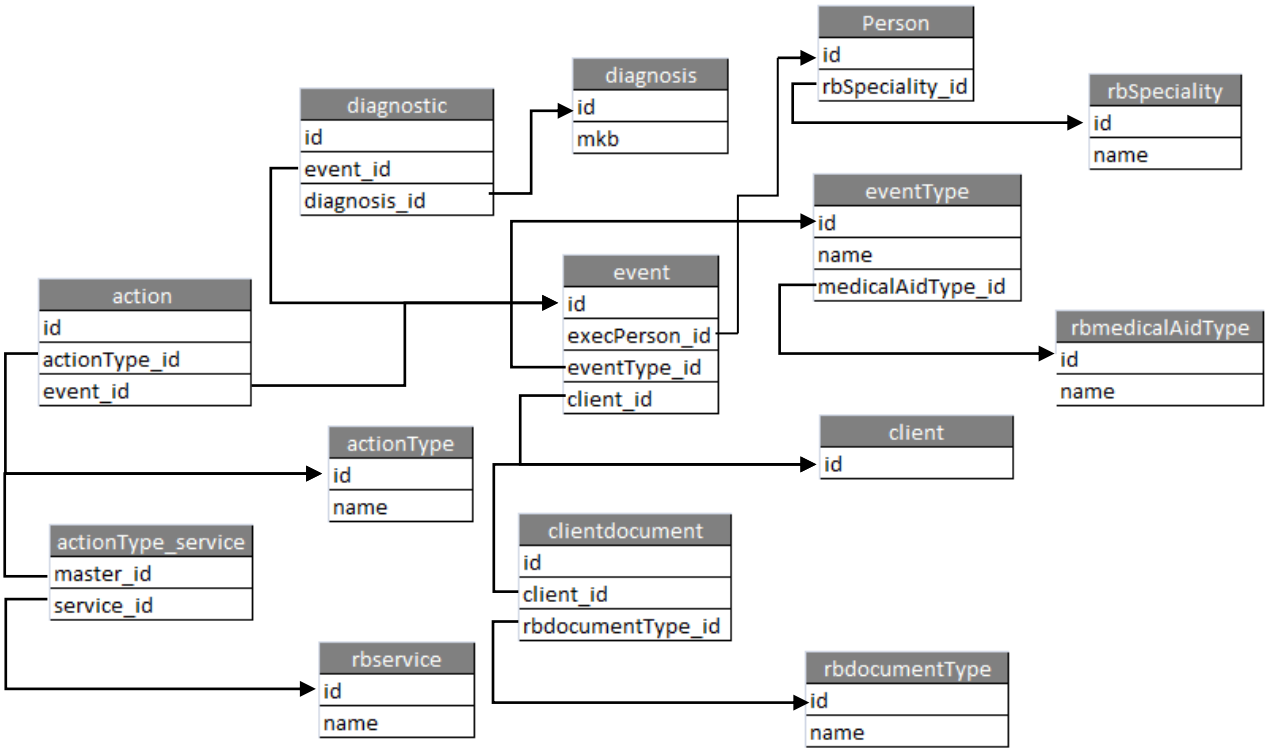
Было замечено, что практически во всех допускаемых ошибках фигурируют следующие поля: медицинская услуга, специальность

сотрудника, диагноз, поставленный пациенту, предоставляемые услуги, возраст и пол пациента, вид оказанной медицинской помощи, вид обращения. Для выявления значений полей, упоминавшихся в справочнике, была изучена структура данной базы.

1.3 Анализ базы данных

Исследовательская работа основывается на базе данных, предоставленной компанией ООО «Виста», содержащей информацию о клиентах, проходивших лечение в ЛПУ. Данная база включает в себя 565 таблиц. При обращении клиента к врачу создается обращение в таблице Event, где указываются идентификаторы медицинского сотрудника, клиента и типа обращения (например, диспансеризация детей, оставшихся без попечения родителей). В свою очередь таблица Client (клиент) прикреплена по внешнему ключу к таблице ClientDocument (документ клиента), где лежат данные о предоставленном документе (например, паспорт гражданина Российской Федерации или свидетельство о рождении). Параллельно в таблице Diagnostic записываются номер обращение (Event) и поставленный диагноз. В таблицу услуг Action (действия) заносятся данные о предоставленных услугах (например, ультразвуковое исследование молочных желез), прикрепленных к обращению. Чтобы подробнее разобраться во внешних ключах между таблицами, была составлена схема, приведенная на рисунке 1, которая включает в себя все нужные для исследования таблицы.

Рисунок 1.Зависимости выбранных таблиц в базе данных



Возвращаясь к списку полей, выявим те, которые часто упоминались в справочнике причин возврата счетов. Таковыми являются медицинская услуга, специальность сотрудника, диагноз, услуги, возраст и пол пациента, вид оказанной медицинской помощи, вид обращения. Для более наглядного примера сопоставления их с наименованиями полей в базе данных, была составлена таблица 2, включающая название полей, их значения, примеры и количество индивидуальных значений, которое принимает каждое поле.

Таблица 2.Отобранные названия полей для дальнейшего исследования

№	Название из справочника	Пример	Название поля в БД	Кол-во уникальных значений
1	Тип действия	Ультразвуковое исследование молочных желез	actionType_name	13673
2	Тип обращения	Профессиональный осмотр	eventType_name	180
3	Название услуги	Флюорография легких	service_name	21835
4	Возраст пациента	Разница между датой рождения и датой обращением	client_age	-
5	Пол пациента	1 - Мужской, 2 - Женский	client_sex	2
6	Диагноз	Расстройства психологического развития	Diagnosis	263
7	Тип документа	Паспорт России, свид. о рож.	documentType_name	63
8	Специальность медицинского сотрудника	Терапевт, хирург	speciality_name	150
9	Вид помощи	Женская консультация, поликлиника взрослая	medicalAidType_name	49

Глава 2. Краткое введение в ассоциативные правила

В данной главе рассматривается понятие и метод поиска ассоциативных правилах (далее - АП). Этот метод известен как анализ рыночной корзины, позволяющий увидеть связи между товарами, найти неочевидные зависимости одного продукта от другого или предсказать покупку комбинации предметов торговли. Когда мы хотим сказать, что два товара зависят друг от друга, мы используем формулировку "если.., то..". Например, если клиент купит кофе, то он купит и молоко. Такие зависимости помогают расставлять товар на полочках в нужные и более привлекательные места для покупателей. Например, в супермаркетах недалеко от полочек с кофе, обычно находятся холодильники с молоком и сливками, а также витрины со сладостями, что провоцирует покупателя на приобретение всех этих трех видов товара.

Задача состоит в том, чтобы применить АП в медицинской документации. Поэтому в качестве товаров будут выступать характеристики электронных медицинских карт или, другими словами, историй болезней разных пациентов.

2.1. Понятие ассоциативных правил

Дана база данных (*dataset*):

$$D = \{C_1, C_2, .., C_n\},$$

где C_j, $j = 1, 2, .., n$ – это электронные карточки пациентов, n – их количество, равное числу карточек, содержащих информацию о взаимосвязанных событиях, из набора D.

Множество возможных характеристик (*itemset*), которые могут входить в истории болезней, обозначим через I:

$$I = \{\alpha_1, \alpha_2, \ldots, \alpha_{N_I}\},$$

где α_k – элементы множества I, $k = 1, \dots, N_I$; $N_I = |I|$ – количество всех элементов в I.

В свою очередь каждый элемент в D, то есть каждая карта пациента, имеет вид:

$$C_j = \{id_j, ch_j\},$$

где id_j – идентификатор C_j карточки, $ch_j = \left\{a_{1,j}, a_{2,j}, \dots, a_{N_{ch_j},j}\right\}$ – набор характеристик истории болезни C_j, где $a_{i,j}$, $i = 1, 2, .., N_{ch_j}$ - i-й элемент из множества ch_j; $N_{ch_j} = \left|ch_j\right|$ – количество элементов множества ch_j. Таким образом, получается, что ch_j – представляет собой набор из элементов множества I.

Определением АП является импликация $X \to Y$, где X и Y - подмножества множества I и $X \cap Y = \emptyset$. Также задача построения АП связана со следующими метриками:

a. Для набор элементов X можно посчитать поддержку($support$) $supp(X)$, которая вычисляется как отношение количества всех карточек, содержащих этот набор, к количеству всех историй болезней в D. Будем говорить, что поддержка правила $X \to Y$ равна поддержке объединения этих множеств:

$$supp(X \to Y) = supp(X \cup Y)$$

Например, пусть поддержка правила "если покупатель приобретет кофе, то он купит и молоко" ($X \to Y$, X – кофе, Y – молоко) равна 25%. Это говорит о том, что в 25% сделанных покупок молоко и кофе ($X \cup Y$) присутствовали одновременно.

b. Введем понятие достоверности ($confidence$) правила. $conf(X \to Y)$ говорит о том, сколько карточек, имеющих в себе набор X, содержат набор Y:

$$conf(X \to Y) = \frac{\text{supp}(\text{X} \cup \text{Y})}{supp(X)}$$

Например, пусть достоверность правила "если покупатель приобретет

кофе, то он купит и молоко" равна 70%. Это говорит о том, что в 70% случаев встречи кофе в списке сделанных покупок присутствует и молоко.

Таким образом, основным результатом поиска АП являются взаимосвязи между характеристиками, для которых также найдены поддержка и достоверность. Тем самым, зная зависимости между признаками в историях болезней, можно будет проверять электронные карты на правильность заполнения.

2.2 Алгоритм поиска ассоциативных правил Apriori

В виду колоссального количества информации о клиентах, хранящегося в МИС, возникает проблема обработки этих данных. Нужен быстрый и эффективный алгоритм, находящий АП и тратящий на это адекватное количество времени.

В 1994г. сотрудником Microsoft Research Ракеш Агравал (*Rakesh Agrawal*) уже была решена эта проблема. Ракеш Агравал придумал основной алгоритм поиска АП - $Apriori$, удовлетворив потребность предприятий розничной торговли в быстром анализе купленных товаров, ведь как только появились штрих-коды, магазины стали сохранять все истории покупок.

На вход алгоритму подается бинарная матрица M, где каждая ее строка это отдельная транзакция:

$$M = \begin{matrix} T_1 \\ T_2 \\ \vdots \end{matrix}$$

где $T_i = \{t_1, t_2, \ldots, t_m\} -$ транзакция, представленная в виде бинарного вектора, в котором $t_j = 1$ говорит о том, что $item_j$ элемент присутствует в T_i; $I = \{item_1, item_2, .., item_n\} -$ множество возможных товаров.

Для примера было рассмотрено множество возможных товаров, состоящее из молока, кофе, печенья, маршмеллоу и сливок. Далее в таблицах приведены примеры обычной базы купленных продуктов и преобразованной по ней бинарной матрицы:

Таблица 3. Обычая база транзакций

Код транзакции	товар 1	товар 2	товар 3	товар 4	товар 5
001	Молоко	Кофе	Маршмеллоу		
010	Кофе	Сливки	Печенье	Маршмеллоу	
100	Кофе	Печенье	Маршмеллоу		
011	Печенье	Молоко	Сливки		
101	Кофе	Молоко	Сливки	Печенье	Маршмеллоу

Таблица 4. Бинарная база транзакций

Код транзакции	Кофе	Маршмеллоу	Молоко	Печенье	Сливки
001	1	1	1	0	0
010	1	1	0	1	1
100	1	1	0	1	0
011	0	0	1	1	1
101	1	1	1	1	1

Получив на вход отсортированную в лексикографическом порядке бинарную базу данных, алгоритм $Apriori$ подсчитывает частоту встреч в транзакциях индивидуальных элементов, затем расширяет их на более крупные наборы. Часто-встречающиеся наборы впоследствии используются для формирования ассоциативных правил.

Самым длительным процессом при вычислении частот является перебор всех элементов для включения их в наборы. Для ускорения работы алгоритм использует свойство анти-монотонности:

$$supp(X) \leq supp(x),$$

где x – набор с минимальной поддержкой из подмножества набора X. Или другими словами можно сказать, что любой набор X будет часто встречающимся, если все его x подмножества тоже часто встречающиеся. Также предварительно задаются значения параметров минимальной поддержки ($minsupport$) и минимальной достоверности ($minconfidence$). Наборы, имеющие значение меньше заданных порогов, рассматриваться алгоритмом изначально не будут по свойству анти-монотонности, что также

ускоряет процесс поиска частоты и АП.

Данный алгоритм можно расписать по следующим шагам:

1. Выявляются все часто встречающиеся одноэлементные наборы.
2. Генерируется список кандидатов - список наборов, которые могут удовлетворить нашим требованиям, то есть могут являться часто встречающимися.
3. По свойству анти-монотонности отсеиваются неподходящие наборы.
4. Отбираются кандидаты, поддержка которых больше или равна заданной $minsupport$.
5. Пункты 2-4 повторяются до тех пор, пока не будут составлены все возможные наборы.

Для наглядности был рассмотрен пример работы алгоритма. Для начала каждому продукту присваивается код, состоящий из одной цифры: кофе - 1, маршмеллоу - 2, молоко - 3, печенье - 4, сливки - 5. Составляется следующая база данных товаров, купленных вместе:

Таблица 5. База транзакций

Транзакции
{1, 2, 3}
{1, 2, 4, 5}
{1, 2, 4}
{1, 3, 4, 5}
{1, 2, 3, 4, 5}

Понятие частоты набора будет использоваться тогда, когда набор элементов появляется, по меньшей мере, 3 раза в рассмотренной базе транзакций: значение 3 будет минимальным значением поддержки.

Первый шаг - подсчет поддержки для каждого элемента. Проанализировав базу, получаем следующее:

Таблица 6. Одноэлементные наборы

Товары	Поддержка
{1}	5
{2}	4
{3}	3
{4}	4
{5}	3

Все товары имеют поддержку больше, либо равное трем, а значит они все частые. На следующем шаге генерируются все пары. Для примера, пара {1, 2} встречается в четырех из всех транзакций, а это значит, что эта пара имеет поддержку четыре.

Таблица 7. Парные наборы

Товары	Поддержка
{1, 2}	4
{1, 3}	3
{1, 4}	4
{1, 5}	3
{2, 3}	2
{2, 4}	3
{2, 5}	2
{3, 4}	2
{3, 5}	2
{4, 5}	3

Пары {2, 3}, {2, 5}, {3, 4} и {3, 5} имеют частоту меньше порога. Отсюда следует, что любое множество, содержащее эти пары, не является частым. После сокращения множества частых наборов, ищутся часто повторяющиеся тройки, не учитывая вышеупомянутые пары:

Таблица 8. Триплеты

Товары	Поддержка
{1, 2, 4}	2
{1, 4, 5}	3
{2, 4, 5}	2

Пара {1, 4, 5} - часто встречающаяся. При составлении четверок все

поддержки наборов меньше минимального значения, как и при пятерках.

После того, как мы были найдены все часто встречающиеся наборы, остается выявить правила, что не является сложной задачей. Чтобы вычислить достоверность правила, будет нужно лишь поделить значение поддержки объединения правой и левой частей правила на поддержку только правой части. Тем самым повторный проход по базе транзакций уже не имеет смысла.

Глава 3. Реализации и получение ассоциативных правил

Все эксперименты проводились на ультрабуке Asus Zenbook UX32LA с процессором Intel Core i7 CPU @ 2,00Ghz и 8 Гб оперативной памяти. В качестве среды разработки использовались интерактивная оболочка для языка Python Jupyter Notebook и R 3.3.3 для языка R. Для работы с базой данных использовались Connector/Python с официального сайта MySQL.

3.1 Инициализация

В первой главе была рассмотрена полезная информация, влияющая на выбор полей для дальнейшей работы. Изучив структуру базы данных и справочника причин возвратов счетов, выбор был сделан в пользу следующих полей, подробное описание которых дано в параграфе 1.3:

actiontype_name, eventtype_name, service_name, client_age, client_sex
diagnosis, documenttype_name, speciality_name, medicalaidtype_name

Все они разбросаны по разным таблицам. Требуется собрать тренировочную выборку. Эта часть исследования была выполнена на языке Python. Работу с базой данных можно разбить на три пункта:

1. Извлечение данных полей из каждой истории болезни
2. Преобразование полученной коллекции в формат, подходящий для инициализации алгоритма поиска ассоциативных правил.
3. Сохранение выборку в csv-файл.

В первом пункте стоит отметить, что для облегчения работы программистов с разными системами управления базами данных была придумана спецификация Database API, скрывающая различия между их реализациями. Так как электронные медицинские карты хранятся в MySQL, поэтому подключаем DB-API модуль *mysql.connector*. Выборка возрастов клиентов выглядит следующим образом:

```
SELECT TIMESTAMPDIFF(year, client.birthDate, action.createDatetime)
FROM action
JOIN event on action.event_id = event.id
JOIN client on event.client_id = client.id
```

Где функция TIMESTAMPDIFF считает количество лет между датой рождения клиента и датой создания обращения, а JOIN подключает нужные таблицы по внешним ключам.

Во втором пункте уделяется внимание отдельным полям, таким как "возраст" и "пол". Для поля возраст годы жизни человека будут разделены на 5 промежутков: от 0 до 1 года, от 2 до 4 лет, от 5 до 17 лет, от 18 до 59 и от 60 и выше. А значение поля "пол" поменяется на строковый тип данных присвоением единице значения Male, двойке - Female.

На последнем этапе полученная выборка была сохранена в *train.csv*. Этот формат предназначен для представления данных в виде таблицы. Поля разделяются запятыми, откуда и название формата - comma-separated values. В итоге обучающая выборка состоит из 1 911 470 строк (транзакций) и девяти столбцов. Ниже приведен пример двух строк из тренировочной выборки:

Таблица 9. Тренировочная выборка

actionType_name	eventType_name	service_name	client_age	client_sex
Исследование уровня глюкозы в крови	(I этап) Профилактические осмотры детей	Исследование уровня глюкозы в крови	0-1	Male
Прием (осмотр, консультация) врача-акушера-гинеколога первичный	32001 Женская консультация	Прием (осмотр, консультация) врача-акушера-гинеколога первичный	18-59	Female

diagnosis	documentType_name	speciality_name	medicalAidType_name
Обращение в учреждения здравоохранения для медицинского осмотра и обследования	СВИД О РОЖД	Педиатр	(I этап) Медицинские осмотры детей
Обращение в учреждения здравоохранения для медицинского осмотра и обследования	ПАСПОРТ РОССИИ	Акушер-гинеколог	Женская консультация

3.2 Пакет *arules* языка R

В данный момент реализацию поиска ассоциативных правил алгоритмом Apriori можно найти в открытом доступе на практически любом языке, но так как наша выборка включает в себя огромное количество строк и полей, то было решено обратить внимание на язык программирования, созданный для обработки данных и работы с графикой - статистический язык R. Он имеет широкое применение в науке, статистике, медицине и экономике. Большим плюсом языка является то, что он поддерживает статистические и численные методы. Благодаря огромному количеству пакетов, R обеспечивает пользователя быстрой добычей данных, их анализа

и иллюстрации. Второе преимущество языка то, что он находится в свободном пользовании, то есть любой человек может скачать и установить его. Также R имеет удобный дизайн, позволяющий пользователю без больших усилий втянуться в работу.

Для поиска ассоциативных правил будем использовать пакет *arules* языка R. Проблема скорости вычисления решена тем, что для сортировки, поиска подмножеств, сопоставления данных и ассоциаций используется разреженное матричное представление, где разреженная матрица - это матрица с преимущественно нулевыми элементами. Еще одним плюсом использования arules можно считать интуитивно понятный интерфейс для анализа и манипулирования данными.

Как говорилось в предыдущей главе, найденные ассоциативные правила должны удовлетворять и минимальной поддержке, и минимальной достоверности. При малом значении поддержки в наборе данных может быть обнаружено большое количество часто встречающихся элементов и ассоциаций, не всегда имеющих весомое значение для базы данных. Практическим решением этой проблемы стало ранжирование найденных правил с помощью дополнительной меры лифт (*lift*):

$$lift(X \to Y) = \frac{supp(X \cup Y)}{supp(X) * supp(Y)} = \frac{conf(X \to Y)}{supp(Y)},$$

Если $lift < 1$, то это говорит о том, что Y является часто встречающимся набором и то, что составленное правило $X \to Y$ не является значимым, т. е. встреча в базе данных набора X не так сильно влияет на встречу Y. Если $lift > 1$, то все наоборот. А равенство лифта единицы означает, что связь между набора X и Y отсутствует.

Для демонстрации значения параметра lift можно рассмотреть пример, добавив в транзакции неподходящий по теме предмет - мыло.

Таблица 10. База транзакций

Код транзакции	товар 1	товар 2	товар 3	товар 4	товар 5
000	Молоко	Кофе	Маршмеллоу		
001	Кофе	Сливки	Печенье	Маршмеллоу	Мыло
010	Кофе	Печенье	Мыло		
011	Печенье	Молоко	Сливки	Мыло	Маршмеллоу
100	Кофе	Молоко	Сливки	Печенье	
101	Мыло				
110	Кофе	Молоко	Мыло	Маршмеллоу	
111	Кофе	Маршмеллоу	Мыло		

$$supp(\text{Кофе}) = \frac{6}{8}, supp(\text{Мыло}) = \frac{6}{8}, supp(\text{Маршмеллоу}) = \frac{4}{8}$$

$$supp(\text{Кофе} \rightarrow \text{Маршмеллоу}) = \frac{4}{8}, supp(\text{Кофе} \rightarrow \text{Мыло}) = \frac{4}{8}$$

$$lift(\text{Кофе} \rightarrow \text{Мыло}) = \frac{8}{9} < 1 \text{ and } lift(\text{Кофе} \rightarrow \text{Маршмеллоу}) = \frac{4}{3} > 1$$

Значение параметра lift больше единицы показывает, что кофе больше влияет на частоту приобретения покупателем маршмеллоу, чем мыла. Даже тот факт, что оно присутствует в шести транзакциях из восьми, в то время как маршмеллоу всего лишь в четырех, не несет никакой полезной информации для ассоциативных правил.

Разобрав структуру входных данных, отметим, что разработанный класс *transactions* эффективно справляется с большим количеством входных разреженных бинарных данных. Матрицу или список можно легко преобразовать в объект класса *transactions.* Значения полей базы данных используются как названия или идентификаторы транзакций. Одной из функций пакета *arules* является реализация алгоритма *Apriori.* Она быстро считает часто встречающиеся наборы, используя поиск в ширину, и находит ассоциативные правила, удовлетворяющие заданным порогам.

3.3 Результаты применения алгоритма

После подключения пакета arules, процесс работы можно разбить на

два этапа:

1. Бинаризация базы данных
2. Построение правил

Считаем обучающую выборку, состоящую из 1 911 470 строк и 9 столбцов. С помощью функции summary(), получим и проанализируем более подробную информацию о базе данных. Например, ниже на рисунке видно, что пациенты женского пола чаще встречаются в базе ЛПУ, чем мужского пола. Это подтверждает тот факт, что женщины чаще болеют, чем мужчины.

Рисунок 2. Количество значений "Male", "Female" в обучающей выборке

```
  client_sex
Male   : 764982
Female:1146488
```

Предварительно числовые показатели, такие как возраст и пол, были сопоставлены соответствующим категориям. Поэтому перейдем к преобразованию базы в матрицу транзакций. После чего в нашей матрице количество строк сохранится - 1 911 470, а вот количество столбцов увеличится до 2570, так как столбцы теперь будут представлять всевозможные бинарные товары (диагноз, пол, тип обращения и т.п). summary() преобразованной в транзакции базы данных дает краткий обзор, показывая самые частые товары, расширенную информацию об элементах (labels), а именно какая переменная (variables) и какое значение (level) использовались для создания каждого двоичного элемента.

Рисунок 3. Результат функции summary()

```
includes extended item information - examples:
                                                                                  labels
1 actiontype_name=|Прием (осмотр, консультация) врача || травматолога-ортопеда первичный|
2                                                    actiontype_name=Абляция эндометрия
3                                                           actiontype_name=Амниоцентез
        variables
1 actiontype_name
2 actiontype_name
3 actiontype_name
                                                                        levels
1 |Прием (осмотр, консультация) врача || травматолога-ортопеда первичный|
2                                                    Абляция эндометрия
3                                                           Амниоцентез
```

Следующим этапом работы будет выявление правил, с помощью функции apriori(). Зададим параметру support значение 0.1 (10%) и confidence - 0.8 (80%). Важно отметить, что все остальные параметры, используемые функцией apriori(), примут значения, заданные по умолчанию. Обратим внимание на параметр maxlen, отвечающий за длину ассоциативного правила. По умолчанию он равен 10, что удовлетворяет нас. Ниже на рисунке представлен обзор построенных правил: после настроек параметров идет описание времени работы выполнения промежуточных действий.

Рисунок 4. Результаты применения функции Apriori()

```
> rul <- apriori (tr, parameter = list (supp = 0.1, conf = 0.8, target = 'rules'))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.8    0.1    1 none FALSE            TRUE       5     0.1      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 191147

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[2570 item(s), 1911470 transaction(s)] done [0.86s].
sorting and recoding items ... [16 item(s)] done [0.08s].
creating transaction tree ... done [2.66s].
checking subsets of size 1 2 3 4 5 6 done [0.00s].
writing ... [252 rule(s)] done [0.00s].
creating S4 object  ... done [0.48s].
```

Результатом алгоритма стало 252 правила при supp = 0.1 и conf = 0.8. Для обзора правил используем также функцию summary(). Она показывает распределение длин часто встречающихся наборов, содержащих в себе элементы и правой, и левой частей правил, то есть, например, найдено 17 часто встречающихся наборов из двух элементов.

Рисунок 5. Результат применения функции summary()

```
> summary(rul)
set of 252 rules

rule length distribution (lhs + rhs):sizes
  2   3   4   5   6
 17  73 102  51   9

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   3.000   4.000   3.849   4.000   6.000

summary of quality measures:
    support         confidence          lift
 Min.   :0.1007   Min.   :0.8087   Min.   :1.427
 1st Qu.:0.1096   1st Qu.:0.9508   1st Qu.:1.700
 Median :0.1377   Median :0.9969   Median :2.761
 Mean   :0.1377   Mean   :0.9676   Mean   :3.183
 3rd Qu.:0.1578   3rd Qu.:1.0000   3rd Qu.:4.456
 Max.   :0.3007   Max.   :1.0000   Max.   :6.185

mining info:
 data ntransactions support confidence
   tr       1911470     0.1        0.8
```

Результаты поиска ассоциативных правил обычно чрезмерно большие, что лишает возможности просмотреть их все. В языке R техники визуализации правил были реализованы в пакете arulesViz. В рамках данной работы был рассмотрен scatter plot (график рассеивания). Для построение графика 1 и графика 2 поддержка и достоверность соответственно равнялись: 0.01 и 0.8 в первом случае (11592 правила); 0.001 и 0.8 во втором (187833 правила), где ось y - значение достоверности, ось x - значение поддержки.

Рисунок 6. График рассеивания 11592 правил

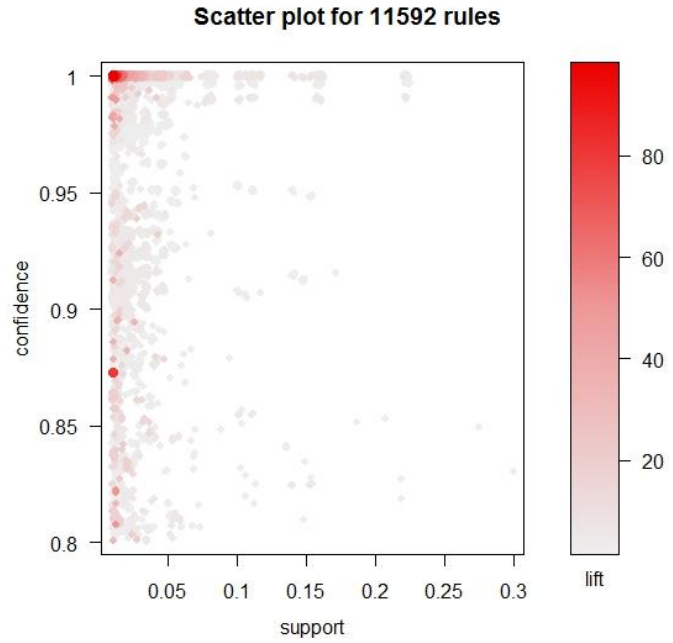
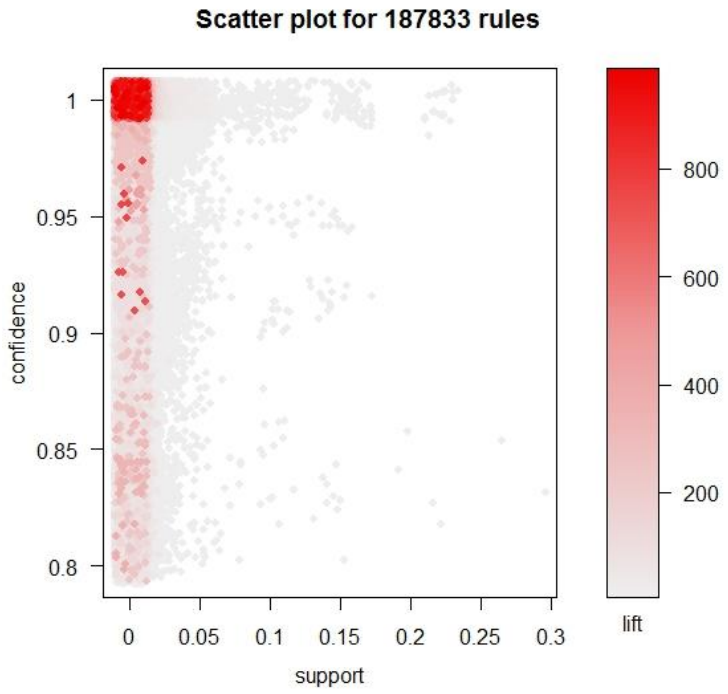


Рисунок 7. График рассеивания 187833 правил



3.4 Ускорение работы функции apriori()

Было найдено 252 правила с поддержкой, равной 10%. Данное найденное количество закономерностей можно считать небольшим числом для базы с 1 911 470 транзакциями, так как 10% - это 191 147 из них. Но при уменьшении порога поддержки скорость вычисления частоты встречаемости наборов и скорость нахождения правил тоже значительно понизятся. Поэтому было решено использовать дополнительную функцию - sample(), которая значительно ускорит процесс поиска правил за счет незначительного ухудшения точности.

Идея сэмплирования (sampling) заключается в том, чтобы выделить случайную выборку и найти в ней все ассоциативные правила, которые также содержатся во всей обучающей выборке. Пусть X - набор входных данных. Нужный размер выборки можно посчитать по формуле:

$$n = \frac{-2 * \ln(c)}{\tau * \varepsilon \hat{} 2},$$

где $\tau = supp(X)$ - поддержка набора X, $A = 1 - \varepsilon$ - точность, ε - в свою очередь относительная погрешность ошибки, $1 - c$ - значение достоверности. Размер выборки не зависит от размера входной базы данных. Для того, чтобы новая выборка включала в себя все частые наборы из входной базы, стоит взять τ, равную порогу $support$, тогда для наборов с поддержкой близкой к минимальному значению, погрешность ошибки сохраняется, в то время как для более частых наборов элементов частота ошибок уменьшается.

Возвращаясь к базе транзакций, в которой 1 911 470 строк и 2570 столбцов, был посчитан приемлемый размер выборки при поддержке равной 10% ($\tau = 0.1$), достоверности 80% ($1 - c = 0.8$) и точности - 90% ($1 - \varepsilon = 0.9$). По формуле получилось $n = 3219$, что значительно меньше размера изначальной выборки. Для сравнения скорости вычисления правил до и после сэмплирования была использована функция system.time(). На рисунках 8 и 9 первый элемент полученного вектора time показывает, сколько времени необходимо ЦПУ (центральное процессорное устройство) для выполнения данного действия.

Рисунок 8. Время работы с dataset

```
> time
пользователь      система       прошло
       27.34         0.30        27.67
```

Рисунок 9. Время работы с sample(dataset)

```
> time
пользователь      система       прошло
        0.08         0.00         0.08
```

Было замечено, что во втором случае, когда сократилось число транзакций, количество правил также уменьшилось на 49. Сопоставив найденные правила в двух разных выборках, было получено 80% сходства, что говорит о том, что практически все частые наборы найдены. Для эксперимента увеличим точность до 99%. Случайная выборка стала больше, а разница между количеством правил сократилась до 21 (92% сходства).

Глава 4. Проверка качества модели

Цель применения ассоциативных правил в медицинской документации - это обнаружение неверно заполненных историй болезней. Для проверки построенной модели на удовлетворение поставленным требованиям была составлена тестовая выборка. За неимением информации о наличии ошибок в медицинских картах, было решено намеренно ввести неверные данные в истории болезней на основе встречающихся ошибок. На языке Python было написано приложение, сравнивающее данные карт из тестовых выборок с правилами, записывающее отчет об ошибках и считающее статистику нахождения несоответствий.

4.1 Программа проверки

Для удобства проверки на сходства карт и правил, все истории болезней были записаны в структуру данных *dict* (словарь), где названия полей стали ключами, а их значения - values. Ниже приведен пример одного элемента из списка словарей:

Рисунок 10. Электронная карта пациента

```
{'actiontype_name': 'Прием (осмотр, консультация) врача травматолога-ортопеда первичный',
 'client_age': '18-59',
 'client_sex': 'Female',
 'diagnosis': 'Обращение в учреждения здравоохранения для медицинского осмотра и обследования',
 'documenttype_name': 'ПАСПОРТ РОССИИ',
 'eventtype_name': '32001 Лечебно-диагностический (поликлиника)',
 'id': '101',
 'medicalaidtype_name': 'Поликлиника взрослая',
 'service_name': 'Прием (осмотр, консультация) врача травматолога-ортопеда первичный',
 'speciality_name': 'Травматолог-ортопед'}
```

Для хранения правил было решено использовать структуру HashMap. которая объединяет в себе возможность доступа к произвольному элементу с динамикой связанного списка. Добавление, удаление или просмотр элемента выполняется за время O(1). Основанная на hash-таблицах, структура HashMap реализует интерфейс Map, позволяя хранить ключи и значения любых типов. С помощью хэш-функции ключ строкового типа преобразуется в значение маленького размера, которое обеспечивает быстрый поиск

нужных элементов. Из этого следует, что подобранная хорошо хэш-функция должна генерировать совершенно разные ключи во избежание совпадений.

При записи в структуру HashMap левая часть найденных правил использовалась как ключ, а правая как значение. Имея такую организацию доступа к ассоциациям, можно легко получить значение, которое следует из запрашиваемого набора данных.

Во время проверки медицинских карточек на соответствие с правилами, составляется отчет об ошибках, куда записываются все те правила, которые были нарушены в данной истории болезни. Параллельно идет подсчет количества найденных корректно неправильных карточек и ошибочно неправильных.

4.2 Эксперименты

В целях исследования оптимальных значений параметров minsupport и minconfidence было решено провести 11 экспериментов, в каждом из которых уменьшался порог поддержки. Последним порогом был 0,005%, что составляет 96 транзакций. Если взять все транзакции и поделить на максимальное количество индивидуальных значений из таблицы 2 (а это количество услуг - 21835), то получим 87 транзакций для одной услуги. Так как услуги неодинаково распределены по транзакциям, то большинство из них попадут в число часто встречающихся наборов.

Таблица 9. Выбранные параметры для 11 экспериментов

	1	2	3	4	5	6	7	8	9	10	11
minsupport	25%	20%	15%	10%	5%	1%	0,5%	0,1%	0,05%	0,01%	0,005%
minconfidence	80%	80%	80%	80%	80%	80%	80%	80%	80%	80%	80%

Для каждого эксперимента в обучающей выборке были найдены ассоциативные правила, быстрое возрастание количества которых показано на графике 1, где под цифрами от 1 до 11 подразумеваются номера экспериментов из таблицы 9 (как и на следующих графиках 2-5).

График 1. Зависимость количества ассоциативных правил от параметров

Для проверки найденных закономерностей было решено составить тестовые выборки из 100 и 1000 транзакций, половина которых неправильно заполненные карточки, а другая половина, соответственно, правильно. Для численной оценки качества найденных правил использовались следующие метрики:

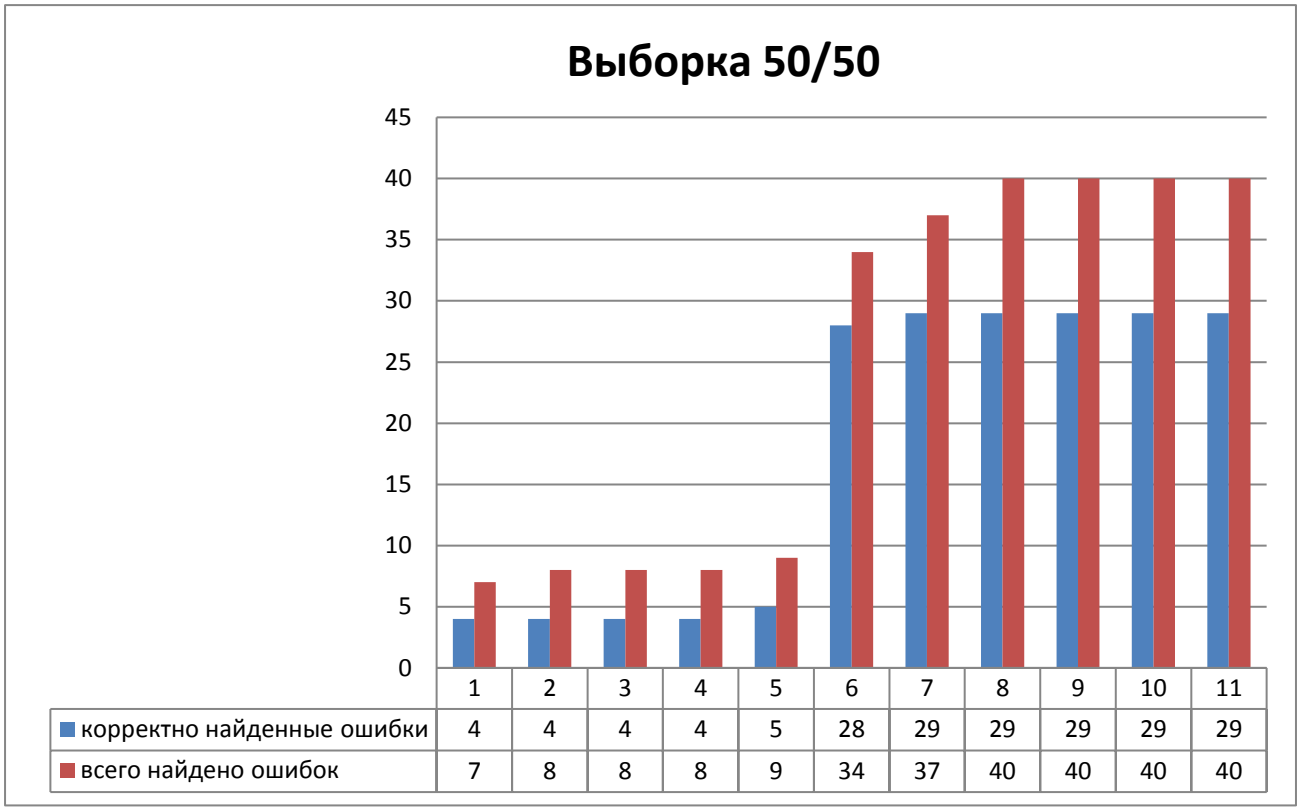
$$precision\ (\text{точноть}) = \frac{\text{количество верно найденных ошибок}}{\text{количество всех найденных ошибок}},$$

$$recall\ (\text{полнота}) = \frac{\text{количество верно найденных ошибок}}{\text{количество ошибок во всей тестовой выборке}},$$

$$F = 2 \times \frac{precision \times recall}{precision + recall},$$

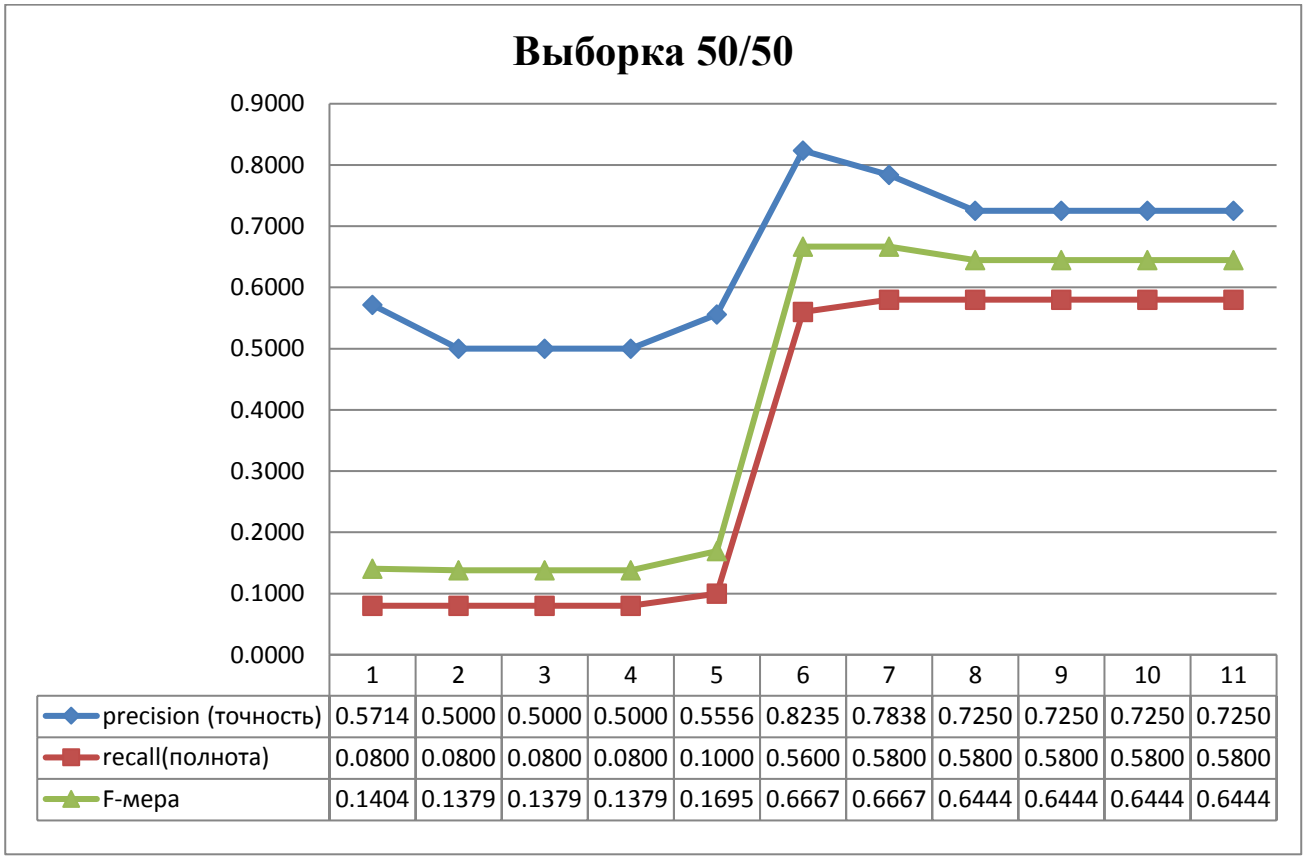
где *precision* показывает, насколько точно найдены ошибки в тестовой выборке, *recall* - сколько ошибок из общего количества обнаружено, а *F-мера* объединяет в себе показатели точности и полноты, когда полнота и точность равнозначны. Для визуального анализа результатов были построены графики количества найденных ошибок и значений метрик.

График 2. Количество всех найденных и верно найденных ошибок в тестовой выборке в каждом эксперименте



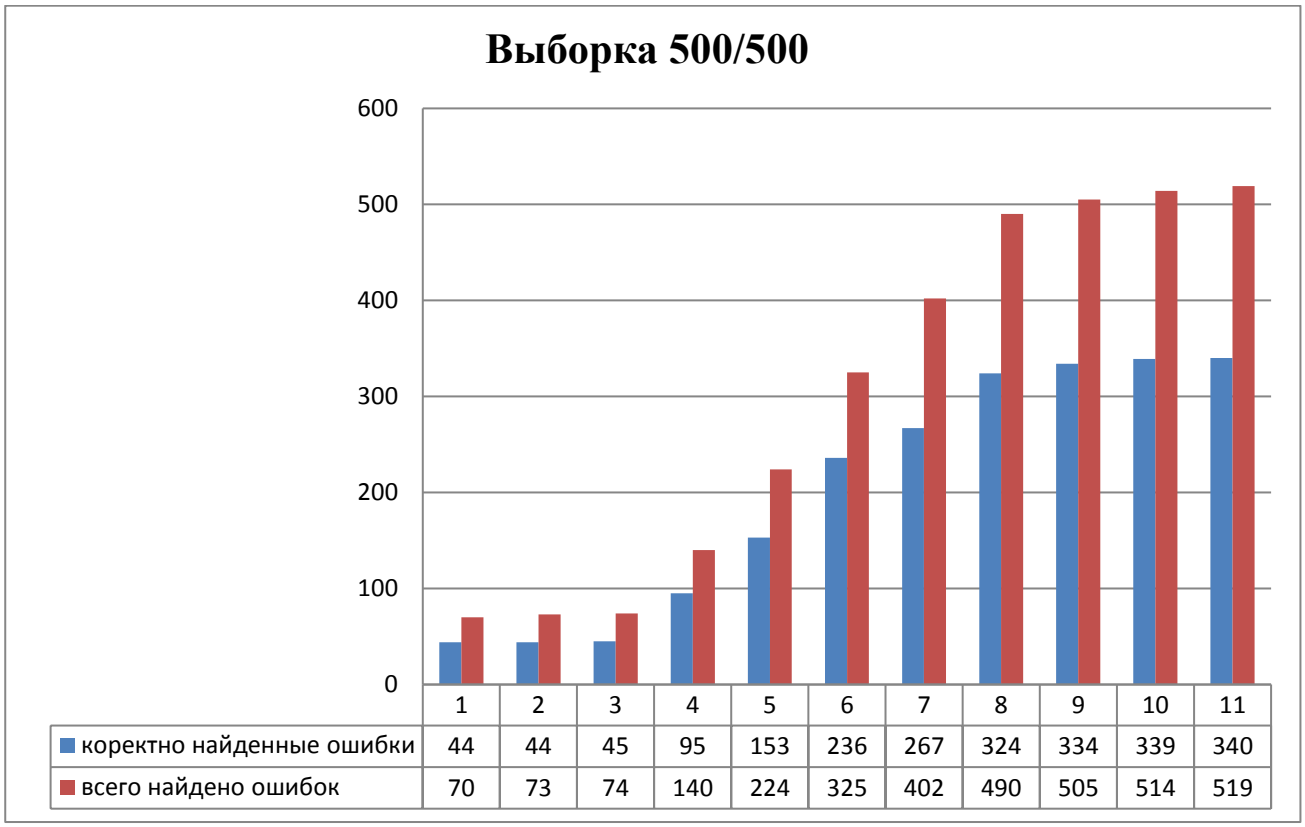
	1	2	3	4	5	6	7	8	9	10	11
корректно найденные ошибки	4	4	4	4	5	28	29	29	29	29	29
всего найдено ошибок	7	8	8	8	9	34	37	40	40	40	40

График 3.Оценки качества ассоциативных правил для каждого эксперимента.



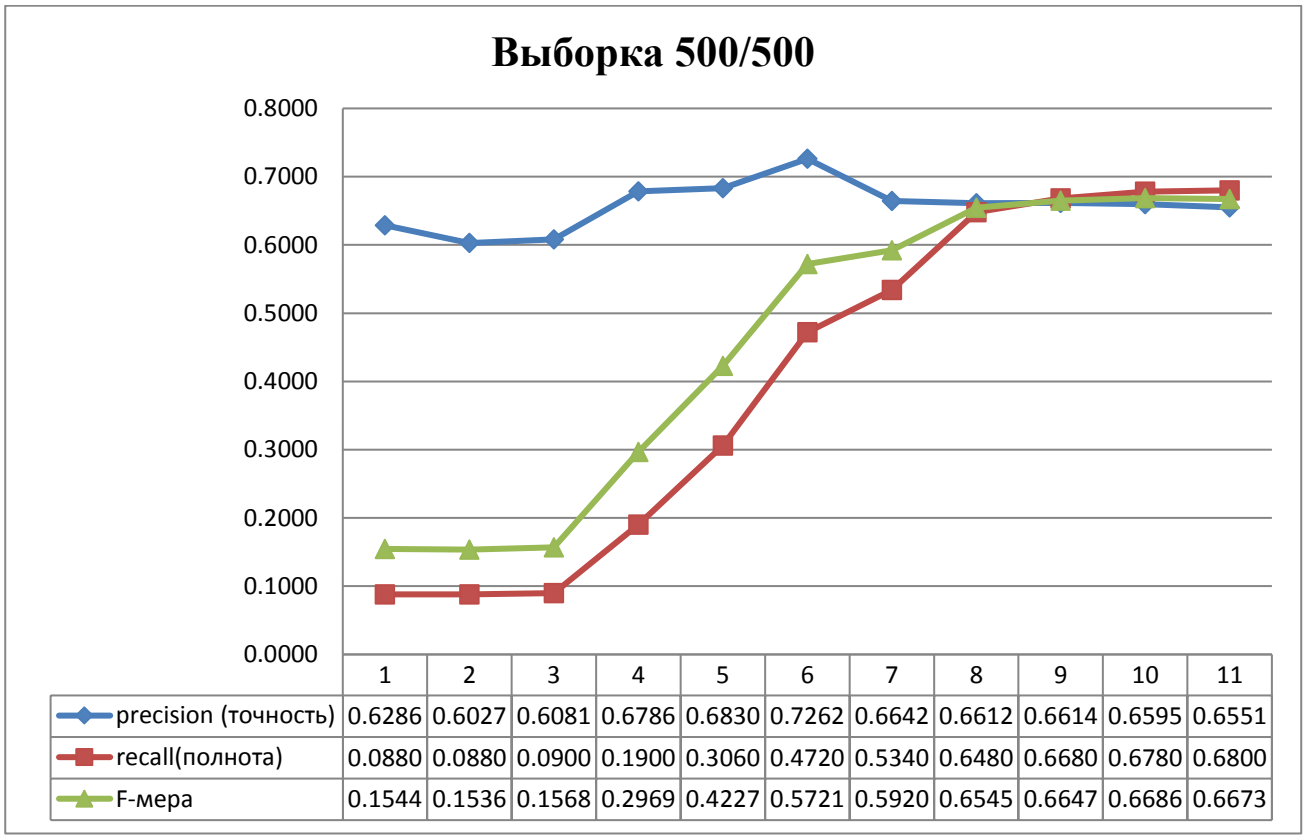
	1	2	3	4	5	6	7	8	9	10	11
precision (точность)	0.5714	0.5000	0.5000	0.5000	0.5556	0.8235	0.7838	0.7250	0.7250	0.7250	0.7250
recall(полнота)	0.0800	0.0800	0.0800	0.0800	0.1000	0.5600	0.5800	0.5800	0.5800	0.5800	0.5800
F-мера	0.1404	0.1379	0.1379	0.1379	0.1695	0.6667	0.6667	0.6444	0.6444	0.6444	0.6444

График 4. Количество всех найденных и верно найденных ошибок в тестовой выборке в каждом эксперименте



	1	2	3	4	5	6	7	8	9	10	11
коректно найденные ошибки	44	44	45	95	153	236	267	324	334	339	340
всего найдено ошибок	70	73	74	140	224	325	402	490	505	514	519

График 5.Оценки качества ассоциативных правил для каждого эксперимента.



	1	2	3	4	5	6	7	8	9	10	11
precision (точность)	0.6286	0.6027	0.6081	0.6786	0.6830	0.7262	0.6642	0.6612	0.6614	0.6595	0.6551
recall(полнота)	0.0880	0.0880	0.0900	0.1900	0.3060	0.4720	0.5340	0.6480	0.6680	0.6780	0.6800
F-мера	0.1544	0.1536	0.1568	0.2969	0.4227	0.5721	0.5920	0.6545	0.6647	0.6686	0.6673

В выборке из 100 транзакций были получены следующие результаты:

- При minsupport равных 1% и 0,5% F-мера достигает максимального значения равного 66% в рамках экспериментов 1-11.
- При уменьшении значения минимальной поддержки F-мера становится меньше, а потом и вовсе перестает меняться, что говорит о ненужности нахождения большего количества правил.
- Значение точности при minsupport равной 1% больше, чем у 0,5%, в то время как полнота меньше. Выбор в пользу одного из значений порога должен быть аргументирован целью применения найденных правил: они должны либо покрывать больше ошибок, либо более точно находить их.

В выборке из 1000 транзакций максимальное значение F-меры равняется 67% при пороге равном 0,01%. Точность в эксперименте с максимальной F-мерой, по сравнению с выборкой из 100 транзакций, стала меньше, а полнота, наоборот, увеличилась.

Для проверки полученных результатов их анализ должен быть проведен специалистами в этой области.

Выводы

В результате выполнения выпускной квалификационной работы можно сделать следующие выводы:

1. После изучения медицинской информационной системы и рассмотрения причин, по которым страховые компании возвращают счета без оплаты, был составлен список полей и их значений, содержащих в себе часто встречающиеся ошибки. Для более глубокого понимания организации хранения электронных медицинских карт пациентов была рассмотрена структура базы данных, а именно значения полей и переходы между таблицами по внешним ключам.

2. Была изучена теория интеллектуального анализа данных, а именно метода построения ассоциативных правил, который далее применялся для обнаружения ошибок в медицинской документации. Разобраны понятия часто встречающихся наборов данных и зависимостей между ними. Показано практическое применение метрик (поддержки и достоверности), характеризующих найденные ассоциативные правила. Рассмотрено нахождение ассоциативных правил алгоритмом Apriori, который на каждом шаге сокращает количество сгенерированных наборов тем, что отсеивает те, которые не прошли по свойству анти-монотонности или были строго меньше минимального значения поддержки.

3. Составлена обучающая выборка, в которой все данные приведены к категориальному типу (например, возраст разбит на интервалы). Освоен пакет arules языка R, в котором важные для исследования функции тратят на подсчет часто встречающихся наборов и на поиск правил гораздо меньше времени, используя разреженные матрицы.

4. Написана программа для проверки медицинских карт на соответствие с обнаруженными ассоциативными правилами. Параллельно приложение ведет статистику о корректности найденных ошибок и качестве

построенной модели.

5. Дополнительно решена задача ускорения поиска ассоциативных правил на больших входных данных сэмплированием матрицы транзакций. При этом экспериментально доказано, что качество найденных ассоциативных правил не ухудшается.

6. Построены графики, отражающие зависимость количества обнаруженных ошибок в историях болезней от значений минимальной поддержки и минимальной достоверности.

7. По графикам можно сделать вывод, что оптимальными значениями минимальной поддержки для выборки из 100 транзакций является 1% или 0,5%, а для 1000 - 0,01%, так как при этих значениях F-мера достигает своего максимального значения.

Заключение

Исследования показали, что поиск ассоциативных правил может быть применим при анализе достоверности медицинской документации. С помощью алгоритма Apriori найденные ассоциативные правила в электронных картах пациентов быстро и эффективно помогают отобрать неправильно заполненные истории болезней, что упрощает работу сотрудников технической поддержки медицинских учреждений. Эффективность и способность работать с большими данными пакета arules показывает, что можно быстро провести анализ огромного количества данных.

Список литературы

1. G. S. Linoff, M. J. A. Berry Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management, 3rd ed., «Wiley», 2011 - 888 p.
2. R. Agrawal, T. Imielinski, A. Swami. Mining association rules between sets of items in large databases. // SIGMOD '93 Proceedings of the 1993 ACM SIGMOD international conference on Management of data, 1993, P 207-216
3. R.Srikant, R. Agrawal, Fast algorithms for Mining Association rules in large database // VLDB '94 Proceedings of the 20th International Conference on Very Large Data Bases, 1994, P 487-499
4. M. Hahsler, S. Chelluboina, K. Hornik, C. Buchta The arules R-Package Ecosystem: Analyzing Interesting Patterns from Large Transaction Data Sets // The Journal of Machine Learning Research, Volume 12, 2/1/2011, P 2021-2025
5. K. Hornik, B. Grün, M. Hahsler. arules - A Computational Environment for Mining Association Rules and Frequent Item Sets // Journal of Statistical Software, 14 (15) ISSN 1548-7660, 2005, P 1-25
6. S. Brin, R. Motwani, J. D. Ullman, S. Tsur. Dynamic Itemset Counting and Implication Rules for Market Basket Data., // SIGMOD '97 Proceedings of the 1997 ACM SIGMOD international conference on Management of data, 1997, P 207-216
7. S. Chelluboina, M. Hahsler. Visualizing Association Rules: Introduction to the R-extension Package arulesViz // Comprehensive R Archive Network, 2010 https://cran.r-project.org/web/packages/arulesViz/vignettes/arulesViz.pdf
8. M. Zaki, S. Parthasarathy, W Li, M. Ogihara, Evaluation of Sampling for Data Mining of Association Rules // Technical report, University of Rochester Rochester, NY, USA ©1996
9. Федеральный закон от 29.11.2010 N 326-ФЗ "Об обязательном медицинском страховании в Российской Федерации".
10. Приказ ФФОМС от 16.04.2012 N 73 "Об утверждении Положений о

контроле за деятельностью страховых медицинских организаций и медицинских организаций в сфере обязательного медицинского страхования территориальными фондами обязательного медицинского страхования".
11. Мухаматзанова М. Ш., Юдин В. А., Карась С. И., Захарова М. А. Об актуальности применения информационных систем в медицине // Медицина и образование в Сибири : электронный журнал, 2007. – № 3.
12. Справочник "Причины возврата счетов" SPR15. http://www.kubanoms.ru/infirmac_obmen1.html?template=print